

Point d'Ã©tape sur Xoops 2.3 : Skalpa allume la mÃ©che

XOOPS

PubliÃ© par : Christian

PubliÃ© le : 18/03/2006



AprÃ©s [la lumiÃ©re au bout](#) suivi de [la feuille de route Xoops 2.3/2.4](#) voici un point intermÃ©diaire sur les fonctionnalitÃ©s de ces futures versions de Xoops.

Comme vous pouvez le lire dans l'Ã©tymologie [article rÃ©digÃ© par Solo](#), l'Ã©tymologie [Afux](#) a rÃ©uni le week-end des 4 et 5 mars les membres de l'Ã©quipe Xoops francophone afin de faire le point sur ce que comportera la future version de Xoops et lui assurer le support qu'il convient auprÃ©s de la communautÃ©.

[Skalpa](#), responsable du dÃ©veloppement du noyau xoops, est l'Ã©tymologie initiateur de cette version. Nous allons tenter de vous faire partager les nouveautÃ©s et quelques principes de fonctionnement de cette version Ã© partir des notes (prises essentiellement par [HervÃ©](#)).

1. PrÃ©-requis

Cette nouvelle version de Xoops nÃ©cessitera au minimum une version 4.3 de PHP, sachant que des nouveautÃ©s ont Ã©tÃ© implÃ©mentÃ©es pour profiter des avantages de PHP5 (voir plus loin au chapitre 4 connexion avec d'Ã©tymologie autres bases de donnÃ©es)

2. Installation

Il sera toujours possible d'Ã©tymologie installer Xoops comme aujourd'hui mais une nouvelle fonction permettra de lancer le noyau en "ligne de commande" afin de l'utiliser pour des tÃ©ches pÃ©riodiques (cron) ou pour pouvoir configurer des fonctions avancÃ©es qui

ne seront pas modifiables en passant par l'interface graphique.

3. DÃ©marrage de Xoops

Xoops disposera de deux modes de fonctionnement : un mode *dÃ©veloppement* utilisÃ© pour la construction, le paramÃ©trage et la mise au point de son site et un mode *production* lorsque celui-ci sera opÃ©rationnel. Un site paramÃ©trÃ© en mode production devrait tourner plus rapidement qu'un site paramÃ©trÃ© en mode dÃ©veloppement car il ne recevra pas tous les messages de debug, de log et les templates seront dÃ©jÃ© compilÃ©s.

Les messages de debug ont changÃ©s, les trois modes de mise au point sont remplacÃ©s par un seul mode debug qui rÃ©sume tout : les erreurs/warning Php et Mysql.

GrossiÃ©rement en mode dÃ©veloppement (oÃ¹ le debug est activÃ© automatiquement) le site utilise le paramÃ©trage contenu dans la base de donnÃ©es, les templates sont recompilÃ©s Ã© chaque utilisation alors qu'en mode production, il gÃ©nÃ©re du code Php (des classes) afin de gagner en vitesse en sollicitant moins la base de donnÃ©es.

Par exemple lorsque les prÃ©fÃ©rences du site sont modifiÃ©es puis appliquÃ©es (dans la base de donnÃ©es), du code Php est gÃ©nÃ©rÃ© directement afin d'Ã©viter de relire les tables plus tard.

Il sera possible de "dÃ©marrer" Xoops en ligne de commande (via le shell du systÃ©me d'exploitation). Une fois le site xoops "dÃ©marrÃ©" par ligne de commande, c'est tout l'environnement Xoops qui sera disponible.

4. Quelques grands principes

Pyro est la librairie (objet) permettant de gÃ©rer l'interface (par exemple les [breadcrumbs](#)). Cette librairie met Ã© disposition des widgets (objets graphiques) un peu comme la librairie graphique QT sous Linux et Windows.

Le systÃ©me de cache est revu. Le cache cÃ© client (mis en place par les navigateurs depuis des annÃ©es) va Ã©tre mieux utilisÃ© et plus intensivement afin d'obtenir des sites plus rapides en affichage. Le cache cÃ© serveur a aussi Ã©tÃ© amÃ©liorÃ©.

Les Ã©crans de redirection ne sont plus les mÃ©mes, ils sont remplacÃ©s.



Le systÃ©me des traductions va Ãªtre totalement revu afin d'utiliser les fichiers [.po](#) bien connus dans le monde Linux. Cela posera moins de problÃ©mes mais il va falloir revoir toutes les traductions. Les fichiers .po seront traitÃ©s et traduits en un format interne.

Le support multilingue est prÃ©vu. D'un point de vue du noyau cela se traduira par l'integration des fonctions du module xlanguage (rÃ©alisÃ© par [phppp](#) qui doit travailler sur le sujet). Par contre cela ne sera pas suffisant et nÃ©cessitera une adaptation des modules par leurs auteurs, afin que ceux-ci exploitent les nouvelles possibilitÃ©s mises Ã leur disposition.

Une nouvelle interface de gestion des bases de donnÃ©es calquÃ©e sur [PDO](#), [nouveau de PHP5](#), a Ã©galement Ã©tÃ© ajoutÃ©e. MÃ©me si son ajout ne permettra pas Ã XOOps de tourner tout de suite sur d'autres bases de donnÃ©es que MySQL, ce choix est un pas dans cette direction. En attendant, cela permettra d'ajÃ© une adaptation rapide des applications utilisant cette interface, donnera aux dÃ©veloppeurs de nouvelles possibilitÃ©s, tout en permettant aux nouveaux venus de se trouver face Ã une couche d'accÃ©s aux bases de donnÃ©es standard et parfaitement documentÃ©e.

L'administration de Xoops ne devrait plus contenir **QUE** ce qui permet de configurer le site et elle va Ãªtre rÃ©organisÃ©e afin que ce soit moins le "bazard".

Le nouveau systÃ©me d'authentification sera organisÃ© sur le principe de services. Il sera alors possible d'ajouter n'importe quel systÃ©me d'authentification : LDAP (bien sur !) mais aussi SGBD, NIS,

Dans la 2.4 la gestion des blocs sera dÃ©portÃ©e dans un module.

4.1 Surcharge

Xoops va utiliser un systÃ©me de surcharge (surcharge basÃ©e sur le concept utilisÃ© en programmation orientÃ©e objet).

4.1.1 Surcharge des thÃ©mes

Il est possible de crÃ©er un thÃ©me qui hÃ©rite d'un autre thÃ©me. Il suffit alors de ne dÃ©crire que les changements !

4.1.2 Surcharge des templates de modules

Dans le thème d'un site (dans son répertoire), il sera possible, en recréant l'arborescence d'un (ou de plusieurs modules), de remplacer les templates par défaut d'un module par des templates personnalisés !

Exemples pratiques

Si tous les sites supports de Xoops (et xoops.org) étaient sur le thème hbergement, on pourrait imaginer d'avoir le thème de xoops.org. Ensuite chaque site de support aurait son propre thème qui "surchargerait" le thème par défaut afin de le personnaliser selon sa culture et son goût, en mettant par exemple son drapeau ou la photo d'un monument qui représente son pays. Il n'aurait donc pas besoin de refaire tout le thème. Ensuite chaque site de support personnaliserait les templates des modules (toujours en les surchargeant) afin de pousser la personnalisation jusqu'au bout.

Comme autre exemple, on peut penser à une chaîne de restauration qui propose un site "global" qui parle de la chaîne mais qui héberge aussi les sites de chaque restaurant. Le site propose un thème et des templates "généraux" qui définissent l'identité visuelle et générale du groupe et chaque restaurant a la possibilité de "personnaliser" son site en surchargeant le thème par défaut qui a été créée par la maison mère afin d'y apporter une touche locale.

Notes : on a quasiment la possibilité de tout surcharger, thèmes, templates, CSS et images.

4.2 Réécriture des URL

Tout d'abord, Il ne faut pas prendre cela uniquement comme une classique réécriture d'url telle que celle que l'on peut mettre en place dans les fichiers .htaccess d'Apache pour faire de l'url-rewriting. Il va être mis en place une gestion globale et totale des [URI](#) (attention, pas [URL](#) mais URI). L'un des objectifs poursuivis est de permettre de faire de la réécriture d'URL (qui au passage devrait aussi fonctionner sous IIS).

Les modules et le noyau ne traiteront plus des url physiques.

Un exemple de la mise en oeuvre pratique de ce système c'est de pouvoir remplacer un module par un autre (ou de le renommer) sans que cela ne pose de problèmes au niveau des URL.

5. Les thèmes

Un thème va se composer en 3 parties :

- Le canevas

- La page
- Le contenu

Le canevas contient les  ments communs au site, par exemple une banniere de pub en haut ou un menu de navigation.

La page est utilis e par et pour le module (on va par exemple y trouver le nom du module ou le nom de la cat gorie en cours ou le "chemin" d'un article)

Le contenu va quant   lui contenir le contenu propre   la page, par exemple un article.

Le fichier qui contiendra le theme ne devra pas obligatoirement s'appeler theme.html (pratique lorsque l'on r cup re par exemple un kit graphique qui n'a pas  t r alis  la base pour Xoops).

Chaque theme va contenir un fichier xo-info.php. Ce fichier va donner des informations sur le theme (un peu comme les fichiers xoops_version.php utilis s actuellement pour les modules).

Un theme peut h riter d'un autre theme qui lui m me h rite d'un autre theme. Une compatibilit  avec les themes existants est n anmoins assur e mais ils ne pourront pas b n ficier des nouveaut s de cette version.

6. Les templates

Le d limiteur Smarty utilis  dans les templates a  t  modifi  afin de permettre aux graphistes de travailler les templates dans des logiciels comme dreamweaver sans que cela ne pose de probl mes.

Les templates ont  t  totalement revus, on a maintenant la possibilit  de g nerer du Pdf ou tout autre format de document bas  sur un vocabulaire xml (comme par exemple des flux RSS ou pourquoi pas des documents OpenOffice).

En fait un template peut contenir les diff rentes versions de sortie d'un m me document. Dans le m me fichier "physique" du template on peut avoir :

- la version  cran d'une news
- la version RSS
- la version PDF
- la version RSS
- etc..

Cela ne pose pas de problÃ©me vis Ã© vis d'une Ã©ventuelle lourdeur d'interprÃ©tation par Smarty car les diffÃ©rentes versions de la page sont "compilÃ©es" et Ã©clatÃ©es en fichiers par le systÃ©me.

Le code renvoyÃ© aux navigateurs sera du **vrai xhtml**.

PossibilitÃ© d'avoir du contenu adaptÃ© aux navigateurs (par exemple pour une mÃªme page, d'avoir une version en tableau pour Internet Explorer et une version full CSS pour les autres.

Il est possible d'intervenir sur le contenu transmis aux templates via un systÃ©me de plugins (un peu comme Smarty et comme dans Mambo).

A terme on devrait pouvoir modifier les templates en mode wysiwyg via un Ã©diteur qui reste Ã© dÃ©velopper. Cet Ã©diteur utilisera le CSS du site afin que le rendu soit le plus proche possible de l'apparence du site.

Dans les templates on n'appellera plus des chemins physiques. Sur le principe, actuellement on peut utiliser XOOPS_UPLOAD_URL dans les templates afin de faire rÃ©fÃ©rence au rÃ©pertoire d'upload de chaque site. En quelque sorte c'est une forme d'abstraction du chemin physique.

Avec cette nouvelle version, il sera possible de faire rÃ©fÃ©rence Ã© un module (dans les templates), en utilisant une notation du genre :
identifiant_module#nom_de_la_page_du_module

Par exemple : `id_news_module#article`

Attention, il ne faut pas confondre le # (et ce qui se trouve aprÃ©s) prÃ©sent dans ce format d'URL avec des ancrages internes Ã© des pages tels que ceux que l'on peut utiliser en html.

C'est le systÃ©me (Xoops) qui se chargera de faire le lien vers l'emplacement physique du module et d'en faire une url rÃ©fÃ©rite (cette fois-ci au sens url rewriting).

7. Les modules

Le dÃ©veloppement de modules va se trouver grandement simplifiÃ© ET accÃ©rÃ©. La crÃ©ation d'un module simple pourrait Ã©tre accessible Ã© un plus grand nombre grÃ¢ce aux nouveautÃ©s fonctionnelles dont disposera cette version. On pourra crÃ©er un module simple 'par exemple' de blog en moins de 5 minutes comme dans Ruby On Rails.

Chaque module se verra dÃ©finir un identifiant et dÃ©finira (entre autre) ses propres pages (avec un id et une description). Ce fichier dÃ©finira aussi les permissions du module.

Par exemple, "ce module est un module pour l'admin uniquement" (ce sera une recommandation, mais si l'administrateur ne change pas les permissions du module, ce sont les permissions dÃ©fÃ©nies par le dÃ©veloppeur dans le module qui seront utilisÃ©es).

Exemple avec le module d'authentification :

```
return Ã© array(
Ã© Ã© Ã© Ã© 'xoBundleIdentifier'Ã© =>Ã© 'mod_xoops_Identification',
Ã© Ã© Ã© Ã© 'xoBundleDisplayName'Ã© =>Ã©
'XOOPSÃ© defaultÃ© identificationÃ© module',
Ã© Ã© Ã© Ã© //xoClassPath'Ã© =>Ã© '/xoops-module.php',

Ã© Ã© Ã© Ã© 'allowFor'Ã© =>Ã© array(Ã© XOOPS_GROUP_ANONYMOUS,Ã©
XOOPS_GROUP_USERSÃ© ),
Ã© Ã© Ã© Ã©
Ã© Ã© Ã© Ã© 'moduleLocations'Ã© =>Ã© array(
Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© 'login'Ã© =>Ã© array(
Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© 'displayName'Ã© =>Ã© 'UserÃ© login',
Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© 'scriptFile'Ã© =>Ã© '/login.php',
Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© 'parameters'Ã© =>Ã© array(
Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© 'login'Ã© =>Ã© array(Ã© Ã© "
,Ã© Ã© XO_TYPE_STRINGÃ© ),
Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© 'password'Ã©
=>Ã© array(Ã© Ã© "Ã© Ã© XO_TYPE_STRINGÃ© ),
Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© 'xoops_redirect'Ã©
=>Ã© array(Ã© Ã© "Ã© Ã© XO_TYPE_STRINGÃ© ),
Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© ),
Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© ),
Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© 'logout'Ã© =>Ã© array(
Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© 'displayName'Ã© =>Ã© 'Logout',
Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© 'scriptFile'Ã© =>Ã© '/logout.php',
Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© 'allowFor'Ã© =>Ã© array(Ã© Ã©
XOOPS_GROUP_USERSÃ© ),
Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© ),
Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© 'lost-password'Ã© =>Ã© array(
Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© 'displayName'Ã© =>Ã© Ã©
'LostÃ© yourÃ© passwordÃ© ?',
Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© 'scriptFile'Ã© =>Ã© '/lostpass.php',
Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© ),
Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© ),
Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© 'xoServices'Ã© =>Ã© array(
Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© 'xoops_identification_LoginForm'Ã© =>Ã© array(
Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© 'xoClassPath'Ã© =>Ã© '/class/loginform.php',

Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© Ã© ),
```

Ã©,Ã© Ã©,Ã© Ã©,Ã© Ã©,Ã©),
);

Les classes Xoopsform permettant de faire des formulaires ont Ã©tÃ© rÃ©visÃ©es, afin d'offrir une meilleure sÃ©paration entre les donnÃ©es, leur manipulation et leur reprÃ©sentation. Cela permettra de remplacer un type de contrÃ»le par un autre en modifiant le template adÃ©quat (les champs de formulaire Ã©tant crÃ©Ã©s dans le template), tout en offrant au passage de notables amÃ©liorations (une validation avancÃ©e des formulaires faite en temps rÃ©el). Dans l'absolu on devrait Ã©galement pouvoir remplacer la gÃ©nÃ©ration des formulaires html classiques par des [xforms](#) par exemple.

Smarty est conservÃ© mais c'est une version modifiÃ©e et corrigÃ©e qui va remplacer la version actuellement utilisÃ©e. Des amÃ©liorations ont Ã©tÃ© apportÃ©es Ã© certaines fonctions de Smarty afin de permettre une gÃ©nÃ©ration plus rapide des pages, et les plug-ins spÃ©cifiques qui vont apparaÃ»tre dans cette version ont tous Ã©tÃ© conÃ©Ã§us comme des extensions du compilateur de templates afin de ne pas avoir d'incidence sur la rapiditÃ©. GrÃ»ce aux plug-ins et aux diffÃ©rentes versions d'une mÃªme page dans un mÃªme template, cela permettra par exemple Ã© chaque module de disposer d'une version RSS de son contenu !

Le moteur de thÃ©mes (qui gÃ©nÃ©re la page "autour" du contenu) est Ã©galement extensible.

On pourrait aussi imaginer d'avoir un plugin qui rÃ©cupÃ©re tout le contenu envoyÃ© aux pages afin de gÃ©nÃ©rer les balises meta. Dans l'absolu cela permet d'Ã©tendre le systÃ©me Ã© l'infini. Le contenu envoyÃ© aux templates est passÃ© aux plug-ins, traitÃ© par ceux-ci puis renvoyÃ© au template.

Les objets de Xoops sont des composants qui contiennent un mini xoops_version.php (xo-version.php).

A terme il n'y aura plus besoin de dÃ©velopper des blocs car le contenu de la page d'un module pourra Ã©tre visualisÃ© sous la forme d'un bloc (et rÃ©ciproquement). Ce qui implique qu'un bloc pourra s'afficher pour une (ou n) page(s) et non plus forcÃ©ment pour toutes les pages d'un module comme aujourd'hui.

Chaque page dÃ©finie dans le fichier xo-version.php donnera une liste des paramÃ©tres qu'elle attend ainsi que leur type. Cela permet d'avoir une protection et une dÃ©sinfection quasi automatique des paramÃ©tres passÃ©s aux pages. La sÃ©curitÃ© se trouvera donc encore renforcÃ©e. Il suffira de passer les paramÃ©tres reÃ©Ã§us par la page Ã© une fonction du noyau afin qu'ils soient nettoyÃ©s et qu'un "cast" (transtypage) soit rÃ©alisÃ©.

Le clonage de modules ne devrait plus Ã©tre nÃ©cessaire.

Deux modules simples comme un module de contact et un module de cr@ation de pages ind@pendantes devraient @tre propos@s avec cette version afin de servir d'exemples pour les d@veloppeurs de modules.

Des normes seront @dict@es pour les d@veloppeurs de modules afin d'aboutir @ une certification de ceux-ci tant en terme de qualit@ que de s@curit@.

8. Multisites

Une gestion complete du multisite est egalement en route. Toutes les nouvelles fonctionalites sont en effet realis@es afin d'etre compatible avec un futur support officiel du multisite (notamment le nouveau gestionnaire de configuration). Meme si ce support necessitera des modifications de vieilles parties du noyau et de nombreux tests avant de se voir qualifi@d "officiel", son utilisation devrait neanmoins etre possible dans un cadre restreint (par des utilisateurs experimentes) des les prochaines versions.

Pour l'utilisateur lambda qui n'utilise qu'un site cela n'a pas forc@ment d'int@r@t au premier abord mais lorsque l'on @alise plusieurs sites cela permet de gagner @norm@ment de temps lors des mises @ jour par exemple.

Cela permet aussi d'avoir 2 versions de son site, une version de production et une version de d@veloppement (dans laquelle sont par exemple visibles les messages de debug).

L'@quivalent du mainfile actuel contiendra un param@trage global par d@faut. Dans ce nouveau "mainfile", on ajoutera le param@trage des autres sites g@r@s par le noyau mais il ne sera n@cessaire de ne d@crire que ce qui change.

Par exemple si le nom du serveur de base de donn@es ne change pas, il ne sera pas n@cessaire de le d@crire pour un site du moment qu'il est d@crit dans la configuration g@n@rale. En fait on va pouvoir "surcharger" la configuration par d@faut pour les diff@rents sites en ne mettant que ce qui change. Les diff@rents sites trait@s par un m@me noyau vont h@riter (comme en programmation objet) d'une configuration par d@faut qu'ils pourront modifier.

Un m@me site pourra donc @tre param@tr@, sur une premi@re adresse, pour ne pas utiliser le mode debug alors que le m@me site mais param@tr@ sur une autre url, lui utilisera le mode debug donc sans que les utilisateurs normaux puissent voir les messages affich@s dans le debug. Le param@trage de ce fichier se fera dans un premier temps manuellement, mais il est pr@vu de d@velopper une interface pour en g@rer le contenu.

9. Redirections

Les redirections seront "matÃ©rialisÃ©es" dans le systÃ©me par une utilisation des sessions. De maniÃ©re pratique, lorsqu'un module aura besoin de faire une redirection, les donnÃ©es de redirection (comme par exemple l'url vers laquelle faire la redirection) seront mises en session, ensuite une VRAIE redirection http sera lancÃ©e puis le systÃ©me utilisera les donnÃ©es mises en session afin de poursuivre la navigation (c'est la mÃ©me technique qu'a utilisÃ© gijoe dans son hack).

On gardera une compatibilitÃ© avec le systÃ©me actuel de redirections mais l'action mise en oeuvre suite Ã la demande de redirection, sera diffÃ©rente.

10. Documentation



Tout ce que rÃ©alise Skalpa est documentÃ© (l'API du noyau notamment) Un gros effort a Ã©tÃ© fait afin de documenter au maximum les choses.

La documentation du noyau a Ã©tÃ© rÃ©alisÃ©e Ã partir d'une version modifiÃ©e de phpdoc pour avoir une documentation de meilleure qualitÃ©.

Le code du noyau ainsi que les fonctionnalitÃ©s peuvent changer tant que le noyau reste en version Alpha. La documentation sera donc publiÃ©e en mÃ©me temps que la version.

Si l'on compare Ã une voiture on pourrait dire que la documentation dÃ©crite ci-dessus concerne les piÃ©ces detachÃ©es et il restera Ã Ã©crire comment rÃ©aliser l'assemblage de ces piÃ©ces.

Conclusion

Point d'tape sur Xoops 2.3 : Skalpa allume la mche

<https://www.frxoops.org/modules/news/article.php?storyid=1962>

Une version alpha 2 de Xoops 2.3 devrait tre publie prochainement pour tre suivie quelques temps aprs de versions beta. Le dlai et le nombre de beta dpendront des bugs trouvs et du temps ncessaire  leur correction.

Les beta versions apportent une stabilit dans les fonctionnalits, seul le code du noyau peut changer pour faire des ajustements ou des corrections de bugs.

Suite  un retard d'un mois, la sortie prvue de la premire beta est prvue pour le mois d'avril 2006

Alors avec tout ce que vous avez lu, ne croyez vous pas que cette version majeure pourrait s'appeler Xoops 3 ?

Skalpa, Herv et Christian